REMARKS/ARGUMENTS

Prior to this amendment, claims 1-26 were pending. In this amendment, no claims are added, amended, or canceled. No new matter is added. Thus, after entry of this amendment, claims 1-26 are pending.

I.   Examiner Interview

On November 19, 2008, a telephonic interview was held between Examiner Syed and the undersigned. During the interview claim 1 was discussed in view of the *Sang* reference. In particular, *Sang* was discussed with regards to the distinction of an object model, which describes the structure of data that is being stored, and a document, such as an XML document that stores data according to an object model. Applicants' representative offered that *Sang* does not describe verifying an object model, but rather describes verifying data stored using an object model that is presumed to be correct. The Examiner indicated that such a distinction would appear to overcome the rejection, and suggested that the present response present arguments describing this distinction for the Examiner's review. Applicants' representative appreciates the Examiners' helpful suggestions and has presented arguments in accordance with the Examiner's suggestions. Applicants thus believe that the claims are allowable over the cited art. Applicants' representative sincerely thanks the Examiner for his time and careful consideration of the arguments presented.

II.   Claim Rejections under 35 USC § 103

Claims 1, 3, 5, 6, 14, 15, 17, 19 and 25-26 are rejected under 35 U.S.C. 103(a) as being obvious over *Drake* (U.S. Publication No. 2003/0070142), in view of *Rasmussen* (U.S. Patent No. 7,185,016), and further in view of Sang-Kyun Kim ("*Sang")* ("Immediate and Partial Validation Mechanism for the Conflict Resolution of Update Operations in XML Databases."). Applicants respectfully submit that these references do not teach or suggest each element of these claims. This rejection is respectfully traversed.

For example, Applicants' claim 1 recites a computer-implemented method of validating metadata for an object model stored in a database, comprising:

> identifying a first subject of validation, wherein the first subject is one of an object, an attribute, an association and a collection of objects;
> determining a context of metadata validation based on the first subject, the context including one of a) the first subject, and b) the first subject and one or more additional subjects;
> determining one or more validation rules for each subject in the context; and
> applying the determined validation rules to each subject in the context,
> **wherein applying the determined validation rules results in one of partially and completely validating the metadata for the object model, a partial validating of the object model allowing an existing portion of the metadata to be validated before all metadata for the object model is determined,**
> **wherein applying the determined validation rules occurs prior to deployment of the object model, a deployment of the object model allowing the object model to be used to store data according to the object model.**

(*emphasis added*). Limitations such as those highlighted above are neither taught nor suggested by these references.

The Office Action admits that the highlighted limitations above are not taught or suggested by *Drake* in view of *Ramussen*. (Office Action Pg. 7-8). Combining *Drake* with *Rasmussen* and *Sang*, even if there were motivation to do so, would still not render obvious Applicants' claim 1. *Sang* teaches a method of immediate and partial validation for conflict resolution of update operations in an XML database (Title, abstract) and is cited as teaching partial or complete validation (Office Action Pg. 7-8). *Sang* teaches a method for updating the content of an XML document (Page 389). A DTD (document type definition) file is parsed to determine the correct format for data elements in the XML document. (Page 389). The data is validated against this DTD that the data element follows the correct format as specified in the DTD. (page 391). If the data element is verified as correctly conforming to the DTD, the data element can be inserted into the XML document without having to validate the entire XML document (partial validation of the XML document). (page 389).

In order to more fully understand why *Sang* does not obviate claim 1, a DTD file and its relationship to an XML document should be understood. A DTD file contains the definition of the data structure that will be used in an XML document. For example, a portion of a DTD file that defines the structure for an entry corresponding to a list of people may be as follows:

DTD File

```
<!ELEMENT people_list (person*)>
<!ELEMENT person (name, gender)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT gender (#PCDATA)>
```

       In this example, the DTD file describes the object model for an XML document. The object model consists of a people_list, which contains multiple data elements of person, as denoted by the person*. A person is defined as an element that contains a name element and a gender element, as denoted by person (name, gender). The name and gender elements are defined as being #PCDATA, which signifies parsed character data, which for purposes of this example can simply be referred to as text information.

       An XML document may be created based on the object model defined in the DTD. A portion of an exemplary XML document based on the above DTD may be as follows:

XML Document using DTD File

```
<people_list>
        <person>
                <name>John</name>
                <gender>Male</gender>
        </person>
        <person>
                <name>Mary</name>
                <gender>Female</gender>
        </person>
        <person>
                <name>David</name>
                <gender>Male</gender>
        </person>
</people_list>
```

       As shown in this example, the XML document contains three entries, John, Mary, and William, who are Male, Female, and Male respectively. The problem being solved as presented in *Sang* occurs when an update to the XML document is performed. For example, a new entry is

inserted into the XML file. Suppose a new entry is inserted into the XML file, but the entry

contains a structural error. For example, the gender and name elements are reversed. Depicted

below is an example of the XML file with an incorrect entry:

XML Document with Error

```
<people_list>
        <person>
                <name>John</name>
                <gender>Male</gender>
        </person>
        <person>
                <name>Mary</name>
                <gender>Female</gender>
        </person>
        <person>
                <name>William</name>
                <gender>Male</gender>
        </person>
        <person>                              ⬅====Inserting new entry
                <gender>Male</gender>          ⬅====ERROR in order
                <name>William</name>           ⬅====ERROR in order
        </person>
</people_list>
```

In *Sang*, the error is detected without having to parse the entire XML file. For example,

parsing the entire file would require starting at the top of the file, and verifying that every entry

conforms to the DTD. The DTD specifies that there may be a person list, which contains

multiple person elements. Each person element contains a name and a gender in that order. The

name and gender are text fields. As should be clear, this scheme is inefficient because the entire

file, not just the updated element, must be parsed.

The disclosure in *Sang* allows for partially validating the XML file by determining a set

of acceptable transitions based on the DTD file. In this simplified example, the only valid

element that may be inserted after <person> is <name>, the only valid entry after <name> is

<gender>, the only valid entry after <gender> is </person>, and the only valid entries after

</person> are either <person> (indicating a new person) or </people_list> indicating the end of the list. Any other element would be an error according to the DTD file. So, in the above example, the system in *Sang* may determine that an item is being inserted after the end of the person entry for William. The valid transitions after the </person> entry for William are either <person> or </person_list>. Because <person> is the next entry, this is acceptable and does not cause an error. However, the only acceptable entry after a <person> element is a <name> element. In the above example, the next element is <gender>. The system in *Sang* is able to detect this is an error by monitoring the transitions without having to parse the entire file.

      Sang does not teach or suggest "*wherein applying the determined validation rules results in one of partially and completely <u>validating the metadata for the object model</u>, a partial validating of the object model allowing an existing portion of the metadata to be validated before all metadata for the object model is determined*" as recited in Applicants' claim 1. As described in the specification, metadata for the object model is the data that defines the object model (e.g. the DTD file), not the data that is stored using the object model (e.g. the XML file). *Sang* begins with the proposition that the DTD, which is data that defines the model of the XML document, already exists and is accurate. No validation, either complete or partial, of the DTD itself is taught by *Sang*. Furthermore, even if the XML file is considered to be the metadata for the object model, it should be clear that *Sang* can not teach or suggest "*a partial validating of the object model allowing an existing portion of the metadata to be validated <u>before all metadata for the object model is determined</u>.*" In order for *Sang* to function at all, all possible element transitions must be known in advance. Without this a priori knowledge, the system in *Sang* would have no way of determining if an improper state transition is an error, or if it has just not been defined yet.

      In addition, *Sang* does not teach or suggest "*wherein applying the determined validation rules occurs <u>prior to deployment</u> of the object model, a deployment of the object model allowing the object model to be used to store data according to the object model.*" Even if validating the data element being inserted is incorrectly interpreted to include validating the DTD or if the XML document is incorrectly interpreted as the object model, *Sang* still does not teach validating the object model <u>prior to deployment</u>. As stated in the claim, an object model cannot be used to

store data until after the object model has been deployed. At best, this incorrect interpretation would teach validating the object model <u>concurrently with deploying</u> the object model.

As such, the combination of *Rasmussen* with *Drake* and *Sang* cannot render obvious Applicants' claim 1, or the claims that depend therefrom. The other claims recite limitations that similarly are not rendered obvious by these references for reasons including those set forth above. Withdrawal of this rejection is respectfully requested.

Claims 2, 4, 18, and 20-21 are rejected under 35 U.S.C. §103(a) as being obvious over *Drake* and *Rasmussen* and *Sang* further in view of *Mikhailov* (US 6,968,500). These claims are not rendered obvious by *Drake* and *Rasmussen* and *Sang* as discussed above. Combining *Mikhailov* with these references, even if there were motivation to do so, still would not render these claims obvious. *Mikhailov* teaches an automatic forms handling system (col. 1, lines 8-15; col. 5, lines 19-38), and is cited as teaching a group of types of associated metadata (Office Action Pg. 16). Even assuming such teaching for sake of argument, a combination of these references still would not result in, or provide motivation for, the application of different validation rules for an object model at different times, such as to allow for partial validation when not all metadata has been determined and complete validation when all metadata has been determined. Additionally, a combination of these references still would not result in, or provide motivation for application of the validation rules prior to the object model being deployed. As such, the combination of *Rasmussen* with *Drake* and *Sang* and *Mikhailov* cannot render obvious these claims. Applicants therefore respectfully request that the rejections with respect to these claims be withdrawn.

Claims 7-13, 16, and 22-24 are rejected under 35 U.S.C. §103(a) as being obvious over *Drake* and *Rasmussen* and *Sang* further in view of *Lindberg* (US 2003/0028540). These claims are not rendered obvious by *Drake* and *Rasmussen* and *Sang* as discussed above. Combining *Lindberg* with these references, even if there were motivation to do so, still would not render these claims obvious. *Lindberg* teaches a system for transferring information between a user interface and a database over a network (paragraph [0010]), and is cited as teaching a first

subject as a root object for a collection of associated objects (Office Action Pg. 18). Even assuming such teaching for sake of argument, a combination of these references still would not result in, or provide motivation for, the application of different validation rules for an object model at different times, such as to allow for partial validation when not all metadata has been determined and complete validation when all metadata has been determined. Additionally, a combination of these references still would not result in, or provide motivation for application of the validation rules prior to the object model being deployed. As such, the combination of *Rasmussen* with *Drake* and *Sang* and *Lindberg* cannot render obvious these claims. Applicants therefore respectfully request that the rejections with respect to these claims be withdrawn.

CONCLUSION

In view of the foregoing, Applicants believe all claims now pending in this Application are in condition for allowance. The issuance of a formal Notice of Allowance at an early date is respectfully requested.

If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at 415-576-0200.

Respectfully submitted,

/Preetam B. Pagar/

Preetam B. Pagar
Reg. No. 57,684

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, Eighth Floor
San Francisco, California 94111-3834
Tel: 415-576-0200
Fax: 415-576-0300
PBP:scz
61547990 v1

OID-2003-093-01